Machine Learning Over Networks, EP3620

Computer Assignment 2

Deterministic-Stochastic Algorithms in Practice

Group 1: Yasaman, Firooz, Ozan

February 21, 2023

1 Main objectives and scope of the assignment

In this assignment, we use logistic regression to model the CO2 emissions in California from the dataset that is taken by [**?**]. In the assignment, we compare the performance of different gradient-based descent methods, mainly: (a) gradient-descent (GD), stochastic gradient descent (SGD), stochastic average gradient descent (SAG) and stochastic variance-reduced gradient (SVRG). We use MATLAB for the simulations and algorithm implementation.

2 Uploading the Data

The dataset is combined of 2921 different data files. Each file contains 4905 measurement data that we will model as X, and 327 simulation data that we will model as Y. The goal of the regression will be obtaining the simulation predictions by the measurements. In this case, we model the dataset as $\mathbf{X} \in \mathbb{R}^{4905\times 2921}$, $\mathbf{Y} \in \mathbb{R}^{327\times 2921}$. In this case, $\mathbf{x}_i \in \mathbb{R}^{4905\times 1}$ and $\mathbf{y}_i \in \mathbb{R}^{327\times 1}$. In this case the weights are modeled as a matrix, $\mathbf{W} \in \mathbb{R}^{4905\times 327}$.

3 Pre-processing and Initialization

3.1 Dataset Preprocessing

Once we analyze the dataset before running the regression algorithms, we realize that the input and output values have a range of 1000. While the smallest number is around 10^{-4}, the largest number is around 1000. Since we use logistic cost function, the large values inside the exponential function can result in infinite in the output. Once we check the distribution of the values, we do not

observe that the large or small values are outliers, since the data was almost uniformly distributed. Therefore, we divide the whole dataset by its standard deviation to normalize the variance of the dataset. So if we say that $\mathbf{A} = [\mathbf{X}, \mathbf{Y}]$, then

$$\mathbf{A}' = \frac{A}{\text{std}(A)}.$$

After normalizing the dataset, we have separated te dataset into the training and test groups. We used 2500 samples for training and 421 samples for the test.

3.2 Initialization of Weights

The other important point is the initialization of the weights. We model the weights with uniform random variables as in $\mathbf{W}_{i,j} \sim U(0, 0.01)$. Note that, smaller weight initialization is also provides lesser initial loss, therefore faster convergence.

4 Gradient of Logistic Ridge Cost

The cost function in the question is defined as

$$f(\mathbf{W}) = \frac{1}{N} \sum_{i \in [N]} f_i(\mathbf{W}) + \lambda \|\mathbf{W}\|_2^2, \tag{1}$$

where $f_i(\mathbf{W}) = \log\left(1 + \exp\left\{-\mathbf{y}_i \mathbf{W}^T \mathbf{x}_i\right\}\right)$. The gradient of this cost function can be described by the following expression

$$\nabla f_{\mathbf{W}} = \frac{1}{N} \sum_{i \in [N]} \nabla f_i(\mathbf{W}) + 2\lambda \mathbf{W}, \tag{2}$$

where the gradient based on a single data sample is

$$\nabla f_i(\mathbf{W}) = \frac{-\mathbf{x}_i y_i}{1 + \exp\left\{\mathbf{y}_i \mathbf{W}^T \mathbf{x}_i\right\}}. \tag{3}$$

In the following, we will utilize the gradient expression to compare the performance of different gradient descent algorithms.

5 Hyper-parameters

For the simulation algorithms, we try different hyper-parameters to tune in. Here are the following observations we get from the different hyper parameter testing:

- While larger λ promotes smaller weights, setting it higher than 0.01 results in divergence for the SGD, SAG methods. It does not effect the GD mainly.
- α is a critical parameter, while setting it higher than 5 would result into high divergence and oscillating cost values, very low step sizes requires many number of iterations.
- ϵ is a critical parameter for SGD since if we choose very high value, although the cost function converges to a some value, it does not find the optimum point as in GD. However, very low ϵ value ends up with very high number of iterations, and computational complexity.

Based on our observations, hyper-parameters given in Table 1 are selected to run the algorithms.

Table 1: Tuned hyper-parameters.

Parameter	Value
α	1
λ	0.01
ϵ	0.001
Iterations	2000

6 Results

6.1 Convergence comparison

Fig 2: two different step sizes

6.2 Computation Duration

In terms of computational complexity, Table 2 provides the time spent to solve each of the algorithms. The results indicate the SGD spends less time computing compared to the other algorithms. The reasons behind this trend are twofold. One is the less number of gradients needs to be updated in each iterations. The other is due to the high memory of the data for GD, all matrix operations also makes the code slower. This can be seen from the SAG, where even though in each iteration only one gradient is updated, the high memory increases the computational time.

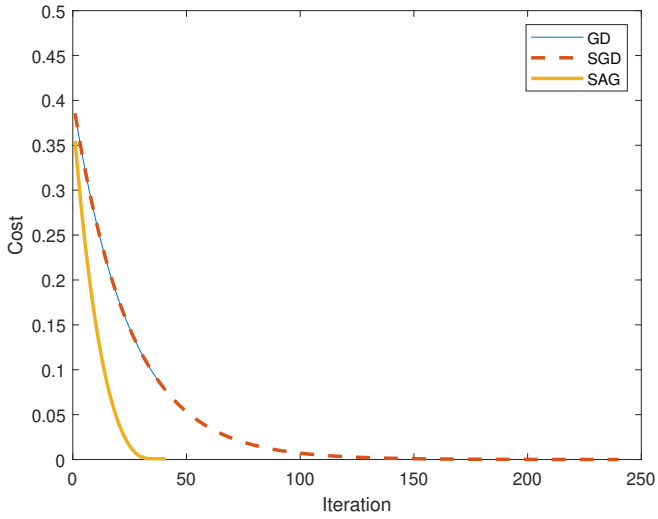

Figure 1: Cost vs iteration for gradient descent algorithms.

Table 2: Comparison of elapsed time.

Algorithm	Total Time (s)	Time per iteration (s)	Number of iterations
GD	1353	19	70
SGD	900	3	300
SAG	2285	17	134

7 Final remarks

We compared different gradient descent algorithms in terms of convergence and computation. The results indicate that SGD method provides good enough convergence with faster computation. For a high size dataset, GD is unnecessarily complex, while for a lower sized dataset, GD can be utilized for faster convergence.